only compute two DFTs and multiply them to filter a section. Letting N_x denote a section's length, the number of computations for a section amounts to

$$(N_x + q)\log_2(N_x + q) + 6(N_x + q).$$

In addition, we must add the filtered outputs together; the number of terms to add corresponds to the excess duration of the output compared with the input (q). The frequency-domain approach thus requires

$$\left(1 + \frac{q}{N_x}\right)\log_2(N_x + q) + 7\frac{q}{N_x} + 6$$

computations per output value. For even modest filter orders, the frequency-domain approach is much faster.

Exercise 5.15.1

Show that as the section length increases, the frequency domain approach becomes increasingly more efficient. Note that the choice of section duration is arbitrary. Once the filter is chosen, we should section so that the required FFT length is precisely a power of two: Choose N_X so that $N_X + q = 2^L$.

Implementing the digital filter shown in the A/D block diagram (Figure 5.24) with a frequency-domain implementation requires some additional signal management not required by time-domain implementations. Conceptually, a real-time, time-domain filter could accept each sample as it becomes available, calculate the difference equation, and produce the output value, all in less than the sampling interval T_S. Frequency-domain approaches don't operate on a sample-by-sample basis; instead, they operate on sections. They filter in real time by producing N_X outputs for the same number of inputs faster than $N_X T_S$. Because they generally take longer to produce an output section than the sampling interval duration, we must filter one section while accepting into memory the **next** section to be filtered. In programming, the operation of building up sections while computing on previous ones is known as **buffering**. Buffering can also be used in time-domain filters as well but isn't required.

Example 5.10

We want to lowpass filter a signal that contains a sinusoid and a significant amount of noise. The example shown in Figure 5.22 shows a portion of the noisy signal's waveform. If it weren't for the overlaid sinusoid, discerning the sine wave in the signal is virtually impossible. One of the primary applications of linear filters is **noise removal**: preserve the signal by matching filter's passband with the signal's spectrum and greatly reduce all other frequency components that may be present in the noisy signal.

A smart Rice engineer has selected a FIR filter having a unit-sample response corresponding a

period-17 sinusoid:

$$h(n) = \frac{1}{17}\left(1 - \cos\left(\frac{2\pi n}{17}\right)\right), \quad n = \{0, \ldots, 16\},$$

which makes $q = 16$. Its frequency response (determined by computing the discrete Fourier transform) is shown in Figure 5.23. To apply, we can select the length of each section so that the frequency-domain filtering approach is maximally efficient: Choose